VHO Prototype Application Programming Interface (API) (8/3/2004)

At present the prototype middleware performs four basic services as well as a querying functionality. All are accessed via SOAP.

Finding information on datasets which are available via the prototype:
datasets()
Inputs – none
Outputs –
Integer, total number of datasets found
String array, each element represents a dataset and contains comma separated Spacecraft Name, Instrument Name and Description

Finding names and descriptions of spatial regions covered by prototype (used later to find data by region):
getregions()
Inputs – none
Outputs –
Integer, total number of regions found
String array, full name and description of region for user followed by short name of region used or finding data by region

Finding data by spatial region:
region(region_name)
Inputs – string – region name (short name, see above)
Outputs -
Integer, total number of datasets that take data in that spatial region
String Array, names of spacecrafts who take data in that spatial region

Querying Data:
query(num_instruments, instruments, start_year, start_doy, stop_year, stop_doy, x1 b1, x2, b2, y1, b3, y2, b4, z1, b5, z2)

Inputs –
num_instruments – integer, number of instruments to send this query too
Instruments – string array, names of instruments to send this query too
Start_year – integer, year to start search at
Start_doy – integer, day of year in start_year to start search at
Stop_year – integer, year to stop search at
Stop_doy – integer, day of year in stop_year to stop search at
X1 – integer, minimum x GSE value to allow
B1 – string, and/or, Boolean one
X2 – integer, maximum x GSE value to allow
B2 – string, and/or, Boolean two
Y1 – integer, minimum y GSE value to allow
B3 – string, and/or, Boolean three

Y2 – integer, maximum y GSE value to allow
B4 – string, and/or, Boolean four
Z1 – integer, minimum z GSE value to allow
B5 – string, and/or, Boolean five
Z2 – integer, maximum z GSE value to allow

Outputs –
Integer, total number of results returned
String array, each element represents a result from an instrument
Values are colon separated – Instrument:ID:Num_Files

Instrument – the name of the instrument that was searched
ID – a unique ID number to associate user and results, used by middleware to keep track of which results correspond to which user during times of heavy load
Num_Files – number of files in this dataset which meet query

Example: To query WIND MFI data for all files in 1999 in which the spacecraft was between 30 GSE X and 40 GSE X

```
query(1, ‘WIND Magnetic Field Investigation’, 1999, 1, 1999, 365, 30, and, 40, and,
          -999, and, 999, and, -999, and, 999)
```

Example: To query WIND MFI data for all files in 1999 in which the spacecraft was either greater than 30 in the GSE X direction or less than –30 in the GSE X direction

```
query(1, ‘WIND Magnetic Field Investigation’, 1999, 1, 1999, 365, -30, or, 30, and,
          -999, and, 999, and, -999, and, 999)
```

The number of files matching the query are returned to the user. However, due to limitations with large SOAP messages the file containing the names and URLs of the data files is left on the data provider computer. The last service provided by the middleware returns to the user a URL for this results file. At this point it is up to the user to obtain this file and any data files listed in it via a method such as Perl LWP. Future versions of the prototype will address this issue by providing the user with software to download these files, however, a solution has not yet been implemented.

Finding the URL of the file that contains the results (filenames and urls) of a query:
download(instrument, id)
Inputs –
Instrument – string, name of Instrument that was searched
ID – the unique ID that was returned by the search
Outputs – URL, string, the url of the results file which contains links to all files which Meet the query